JMLR: Workshop and Conference Proceedings 29:293–308, 2013 ACML 2013

Generalized Aitchison Embeddings for Histograms

Tam Le TAM.LE@IIP.IST.I.KYOTO-U.AC.JP
Marco Cuturi MCUTURI@I.KYOTO-U.AC.JP
Graduate School of Informatics, Kyoto University.

Editor: Cheng Soon Ong and Tu Bao Ho

Abstract

Learning distances that are specifically designed to compare histograms in the probability simplex has recently attracted the attention of the community. Learning such distances is important because most machine learning problems involve bags of features rather than simple vectors. Ample empirical evidence suggests that the Euclidean distance in general and Mahalanobis metric learning in particular may not be suitable to quantify distances between points in the simplex. We propose in this paper a new contribution to address this problem by generalizing a family of embeddings proposed by Aitchison (1982) to map the probability simplex onto a suitable Euclidean space. We provide algorithms to estimate the parameters of such maps, and show that these algorithms lead to representations that outperform alternative approaches to compare histograms.

Keywords: metric learning for histograms, Aitchison geometry

1. Introduction

Defining a distance to compare objects of interest is an important problem in machine learning. Many metric learning algorithms were proposed to tackle this problem in a supervised way, using Mahalanobis distances as a template (Xing et al., 2002; Schultz and Joachims, 2003; Kwok and Tsang, 2003; Goldberger et al., 2004; Shalev-Shwartz et al., 2004; Globerson and Roweis, 2005). In particular, it seems that the Large Margin Nearest Neighbors approach of Weinberger et al. (2006; 2008) and Information-Theoretic Metric Learning by Davis et al. (2007) have emerged as popular tools to learn such metrics.

Among such objects of interest, histograms – the normalized representation for bags of features – play a fundamental role in many applications, from computer vision (Julesz, 1981; Perronnin et al., 2010; Vedaldi and Zisserman, 2012), natural language processing (Salton and McGill, 1983; Joachims, 2002; Blei et al., 2003; Blei and Lafferty, 2006), speech processing (Doddington, 2001; Campbell et al., 2003) to bioinformatics (Burge et al., 1992; Leslie et al., 2002). Mahalanobis distances can of course be used as such on histograms or bags-of-features, but fail to incorporate the geometrical constraints of the probability simplex (non-negativity, normalization) in their definition. Given this issue, Cuturi and Avis (2011) and Kedem et al. (2012) have very recently proposed to learn the parameters of distances specifically designed for histograms, namely the transportation distance and a generalized variant of the χ^2 distance respectively.

We propose in this work a new approach to compare histograms, which builds upon older work by Aitchison (1982). In a series of influential papers and monographs, Aitchison

(1980; 1982; 1985; 1986; 2003) proposed to study different maps from the probability simplex onto a Euclidean space of suitable dimension. These maps are constructed such that they preserve the geometric characteristics of histograms yet make subsequent analysis easier by relying only upon Euclidean tools, such as Euclidean distances, quadratic forms and ellipses. Our goal in this paper is to follow this line of work and propose suitable maps from the probability simplex to $\mathbb{R}^d$ before carrying out classical Mahalanobis metric learning. However, rather than relying on a few mappings defined a priori, such as those proposed in (Aitchison, 1982), we propose to *learn* such maps directly in a supervised fashion.

This paper is organized as follows: after providing some background on Aitchison transformations in Section 2, we propose a generalization of Aitchison embeddings in Section 3. In Section 4, we propose an algorithm to learn the parameters of such embeddings using training data. We also review related work in Section 5, before providing experimental evidence in Section 6 that our approach improves upon other adaptive metrics on the probability simplex. We conclude in Section 7.

2. Aitchison Transformations

We consider the probability simplex of d-dimensional histograms,

$$\mathbb{S}^d \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^d \;\middle|\; \sum_{i=1}^{d} \mathbf{x}_i = 1 \text{ and } \mathbf{x}_i \geq 0, 1 \leq i \leq d \right\},$$

throughout this paper. Aitchison (1982, 1986, 2003) claims that the information reflected in histograms lies in *the relative values of their components* rather than on their absolute value. Therefore, using a Euclidean distance between histograms is not appropriate, since it only considers arithmetic differences between components. To tackle this issue, Aitchison proposed to define several embeddings for histograms that stress the importance of ratios. A central element in Aitchison's analysis is the log-ratio between components of a vector $\mathbf{x}$,

$$\log \frac{\mathbf{x}_i}{\mathbf{x}_j} = \log \mathbf{x}_i - \log \mathbf{x}_j,$$

which appears throughout the three embeddings proposed in his work, which we detail below.

2.1. Additive log-ratio transformation

The first transformation proposed by Aitchison (1982, p.144, 2003, p.29) is the additive log-ratio transformation (**alr**) which maps a vector $\mathbf{x}$ from the probability simplex $\mathbb{S}^d$ onto $\mathbb{R}^{d-1}$,

$$\mathbf{alr}\,(\mathbf{x}) \stackrel{\text{def}}{=} \begin{bmatrix} \vdots \\ \log \frac{\mathbf{x}_i + \varepsilon}{\mathbf{x}_d + \varepsilon} \\ \vdots \end{bmatrix}_{1 \leq i \leq d-1} \in \mathbb{R}^{d-1},$$

where ε is a small positive number. The **alr** map for $\mathbf{x} \in \mathbb{S}^d$ can be reformulated as:

$$\mathbf{alr}\left(\mathbf{x}\right) = \mathbf{U}\log\left(\mathbf{x}+\varepsilon\mathbf{1}_d\right), \mathbf{U} = \begin{bmatrix} 1 & \cdots & 0 & -1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -1 \end{bmatrix}, \tag{1}$$

where $\mathbf{U} \in \mathbb{R}^{(d-1)\times d}$, $\mathbf{1}_d \in \mathbb{R}^d$ is the vector of ones, and $\log \mathbf{x}$ is the element-wise logarithm.

The formula of the **alr** transformation is directly derived from the definition of the logistic-normal distribution (Aitchison and Shen, 1980; Blei and Lafferty, 2006) on $\mathbb{S}^d$, which is simply equivalent to a multivariate normal distribution on the image of the **alr** transformation onto $\mathbb{R}^{d-1}$. The **alr** map is an isomorphism between $\left(\mathbb{S}^d, \oplus, \otimes\right)$ and $\left(\mathbb{R}^{d-1}, +, \times\right)$ where $\oplus$ and $\otimes$ are the perturbation (Aitchison, 2003, p.24) and power (Aitchison, 2003, p.26) operations respectively in the probability simplex, but not isometric since it does not preserve the distance between them.

2.2. Centered log-ratio transformation

The second transformation proposed by Aitchison (2003, p.30) is the centered log-ratio transformation (**clr**), which considers now the log-ratio of each coordinate of $\mathbf{x}$ with the geometric mean of all coordinates,

$$\mathbf{clr}\left(\mathbf{x}\right) \stackrel{\text{def}}{=} \left[\log \frac{\vdots}{\frac{\mathbf{x}_i+\varepsilon}{\sqrt[d]{\prod_{j=1}^{d}\left(\mathbf{x}_j+\varepsilon\right)}}} \vdots\right]_{1\leq i\leq d} \in \mathbb{R}^d. \tag{2}$$

The **clr** map can be also expressed with simpler notations using a weight matrix and the element-wise logarithm:

$$\mathbf{clr}\left(\mathbf{x}\right) = \left(\mathbf{I} - \frac{\mathbf{1}_{d\times d}}{d}\right)\log\left(\mathbf{x}+\varepsilon\mathbf{1}_d\right).$$

Here, $\mathbf{I}$ and $\mathbf{1}_{d\times d}$ stand for the identity matrix and the matrix of ones in $\mathbb{R}^{d\times d}$. The **clr** map is not only an isomorphism, but also an isometry between the probability simplex $\mathbb{S}^d$ and $\mathbb{R}^d$. Note that the **clr** map spans the orthogonal of $\mathbf{1}_d$ in $\mathbb{R}^d$.

2.3. Isometric log-ratio transformation

Egozcue et al. (2003) proposed to project the images of the **clr** map onto $\mathbb{R}^{d-1}$, to define the *isometric log-ratio transformation* (**ilr**). The **ilr** map is defined as follows:

$$\mathbf{ilr}\left(\mathbf{x}\right) \stackrel{\text{def}}{=} \mathbf{V}\mathbf{clr}\left(\mathbf{x}\right) = \mathbf{V}\left(\mathbf{I} - \frac{\mathbf{1}_{d\times d}}{d}\right)\log\left(\mathbf{x}+\varepsilon\mathbf{1}_d\right), \tag{3}$$

where $\mathbf{V} \in \mathbb{R}^{(d-1)\times d}$, whose row vectors describe a base of the null space of $\mathbf{1}_d^T$ in $\mathbb{R}^d$. The **ilr** transformation maps a histogram in the probability simplex $\mathbb{S}^d$ onto $\mathbb{R}^{d-1}$, and is also an isometric map between both spaces in Aitchison's sense.

Remark 1 *Aitchison's original definitions do not consider explicitly the regularization coefficient ε (1982; 1986; 2003). In that literature, the histograms are either assumed to have strictly positive values or the problem is dismissed by stating that all values can be regularized by a very small constant (Aitchison, 1985, p.132; 1986, §11.5). We include explicitly this constant ε here because it forms the basis of the embeddings we propose in the next section.*

3. Generalized Aitchison Embeddings

Rather than settling for a particular weight matrix – such as those defined in Equations (1), (2) or (3) – and defining a regularization constant ε arbitrarily, we introduce in the definition below a family of mappings that leverage instead these parameters to define a flexible generalization of Aitchison's maps.

Definition 2 *Let $\mathbf{P}$ be a matrix in $\mathbb{R}^{m\times d}$ and $\mathbf{b}$ be a vector in the strictly positive orthant $\mathbb{R}^d_+$. We define the generalized Aitchison embedding $\mathfrak{a}(\mathbf{x})$ of a point $\mathbf{x}$ in $\mathbb{S}^d$ parameterized by $\mathbf{P}$ and $\mathbf{b}$ as*

$$\mathfrak{a}(\mathbf{x}) \stackrel{\text{def}}{=} \mathbf{P}\log(\mathbf{x}+\mathbf{b}) \in \mathbb{R}^m. \tag{4}$$

Vector $\mathbf{b}$ in Equation (4), can be interpreted as a pseudo-count vector that weights the importance of each coordinate (or bin) of $\mathbf{x}$. Figure 1 illustrates the influence of the pseudo-count vector $\mathbf{b}$ on each coordinate. A large value for $\mathbf{b}_i$ directly implies that the map for that value is nearly constant, thereby canceling the influence of that coordinate in subsequent analysis. Smaller values for $\mathbf{b}_i$ denote on the contrary influential coordinates, or bins.

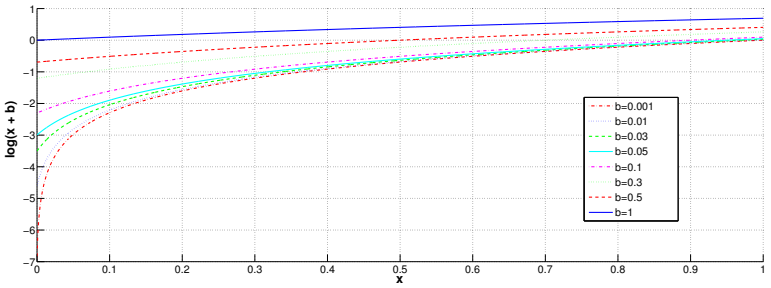

Figure 1: Influence of different pseudo-count values in the logarithm function.

We propose to *learn* $\mathbf{P}$ and $\mathbf{b}$ such that histograms mapped following $\mathfrak{a}$ can be efficiently discriminated using the Euclidean distance. The Euclidean distance between the images of two histograms $\mathbf{x}$ and $\mathbf{z}$ under the embedding $\mathfrak{a}$ is:

$$d_{\mathfrak{a}}(\mathbf{x},\mathbf{z}) \stackrel{\text{def}}{=} d\left(\mathfrak{a}(\mathbf{x}),\mathfrak{a}(\mathbf{z})\right) = \left\|\mathbf{P}\log(\mathbf{x}+\mathbf{b}) - \mathbf{P}\log(\mathbf{z}+\mathbf{b})\right\|_2 = \left\|\log\left(\frac{\mathbf{x}+\mathbf{b}}{\mathbf{z}+\mathbf{b}}\right)\right\|_{\mathbf{P}^T\mathbf{P}}, \tag{5}$$

where the division between two vectors is here considered element-wise. We will also consider the quadratic form $\mathbf{Q} = \mathbf{P}^T\mathbf{P}$ associated with $\mathbf{P}$. Let $\mathcal{S}^+ \stackrel{\text{def}}{=} \left\{\mathbf{Q} \;\middle|\; \mathbf{x}^T\mathbf{Q}\mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^d\right\}$ be a set of positive semi-definite matrix, so $\mathbf{Q} \in \mathcal{S}^+$ or abbreviated as $\mathbf{Q} \succeq 0$. Our goal is to learn either $\mathbf{P}$ or $\mathbf{Q}$ in addition to the pseudo-count vector $\mathbf{b}$ to obtain an embedding that performs well with k-nearest neighbors.

4. Learning Generalized Aitchison Embeddings

4.1. Criterion

We follow Weinberger et al.'s approach to define a criterion to optimize the parameters $(\mathbf{Q}, \mathbf{b})$ of generalized Aitchison embeddings (2006). Ideally, a distance should be parameterized following the information contained in a subset, where the k-nearest neighbors (or target neighbors) of each point should belong to the same class, whereas other points from different classes should be sufficiently far (large margin). Consequently, we consider the following problem for a training set $(\mathbf{x_i}, y_i)_i$:

$$\begin{aligned} \min_{\mathbf{Q},\mathbf{b}} \quad & \mathcal{F} \stackrel{\text{def}}{=} \sum_{i,j \rightsquigarrow i} d^2_{\mathfrak{a}}(\mathbf{x_i}, \mathbf{x_j}) + \mu \sum_{i,j \rightsquigarrow i} \sum_{\ell} (1 - y_{i\ell}) \mathcal{H}_{ij\ell} + \lambda \|\mathbf{b}\|_2^2 \\ \text{s.t.} \quad & \mathbf{Q} \succeq 0, \, \mathbf{b} > \mathbf{0}_d, \end{aligned} \tag{6}$$

where $\mathcal{H}_{ij\ell} \stackrel{\text{def}}{=} \left[1 + d^2_{\mathfrak{a}}(\mathbf{x_i}, \mathbf{x_j}) - d^2_{\mathfrak{a}}(\mathbf{x_i}, \mathbf{x}_\ell)\right]_+$, $j \rightsquigarrow i$ means that the vector $\mathbf{x_j}$ is in the target neighbors of $\mathbf{x_i}$. This notation is not symmetric since $i \rightsquigarrow j$ does not necessarily imply that $j \rightsquigarrow i$. $\|.\|_2$ denotes the ℓ_2-norm. y_i is the corresponding label of a point $\mathbf{x_i}$, and $y_{i\ell}$ is an indicator variable such that $y_{i\ell} = 1$ if $y_i = y_\ell$, and $y_{i\ell} = 0$ otherwise. μ is a trade-off constant of the second objective term in $\mathcal{F}$, and λ is a constant number to penalize for ℓ_2-regularization of the pseudo-count vector $\mathbf{b}$. $[t]_+ \stackrel{\text{def}}{=} \max(t, 0)$ is the hinge loss function.

In the optimization, the first objective term penalizes large distances between each vector $\mathbf{x_i}$ and its target neighbors $\mathbf{x_j}$ while the second objective term penalizes small distances between points which have different labels, and the last term in the objective is an ℓ_2-regularization for the pseudo-count vector $\mathbf{b}$. As analyzed in Section 3, the constraint $\mathbf{Q} \succeq 0$ ensures that we learn a well-defined pseudo-metric.

4.2. Alternating Optimization

We propose a naive approach to learn parameters $(\mathbf{Q}, \mathbf{b})$ of the non-convex optimization (6) by an alternating optimization scheme as described in Algorithm 1. We compute target neighbors for each point using a naive k-nearest neighbors search with $d_{\mathfrak{a}}$ (Equation (5)) as a base distance.

4.2.1. Fix the pseudo-count vector $\mathbf{b}$ to learn the matrix $\mathbf{Q}$

With a fixed vector $\mathbf{b}$, we show that optimization problem (6) with respect to $\mathbf{Q}$ can be cast as a Mahalanobis metric learning problem. By mapping each training vector $\mathbf{x}$ onto $\log(\mathbf{x} + \mathbf{b})$, problem (6) is equivalent to the LMNN optimization problem (Weinberger et al., 2006) where the training data are mapped vectors $\log(\mathbf{x} + \mathbf{b})$ and corresponding labels y.

4.2.2. Fix the matrix $\mathbf{Q}$ to learn the pseudo-count vector $\mathbf{b}$

With a fixed matrix $\mathbf{Q}$, we use a projected subgradient descent to learn the pseudo-count vector $\mathbf{b}$. Defining $g(\mathbf{b}; \mathbf{x_i}, \mathbf{x_j}) \stackrel{\text{def}}{=} d^2_{\mathfrak{a}}(\mathbf{x_i}, \mathbf{x_j})$ we can compute the gradient of g as:

$$\nabla g(\mathbf{b}; \mathbf{x_i}, \mathbf{x_j}) = 2 \left(\mathbf{Q} \log \frac{\mathbf{x_i} + \mathbf{b}}{\mathbf{x_j} + \mathbf{b}} \right) \circ \left(\frac{1}{\mathbf{x_i} + \mathbf{b}} - \frac{1}{\mathbf{x_j} + \mathbf{b}} \right),$$

Algorithm 1 Alternating optimization (AO) to learn parameters of the generalized Aitchison embedding

Input: data $(\mathbf{x_i}, y_i)_i$, a pseudo-count vector $\mathbf{b}_0$ and a matrix $\mathbf{Q}_0$.
Set $t \leftarrow 0$.
repeat
  Find target neighbors for each vector $\mathbf{x_i}$ with $d_{\mathfrak{a}}$ as in Equation (5) at $(\mathbf{Q}_t, \mathbf{b}_t)$.
  Calculate $\mathbf{Q}_{t+1} \leftarrow$ LMNN algorithm with training data $(\log(\mathbf{x_i} + \mathbf{b}_t), y_i)_i$ and an initial matrix $\mathbf{Q}_t$.
  Update target neighbors for each vector $\mathbf{x_i}$ at $(\mathbf{Q}_{t+1}, \mathbf{b}_t)$.
  Calculate $\mathbf{b}_{t+1} \leftarrow$ Algorithm 2 with training data $(\mathbf{x_i}, y_i)_i$, matrix $\mathbf{Q}_{t+1}$ and an initial vector $\mathbf{b}_t$.
  Calculate $\mathcal{F}_{t+1} \leftarrow \mathcal{F}(\mathbf{Q}_{t+1}, \mathbf{b}_{t+1})$.
  $t \leftarrow t + 1$.
until $t < t_{\max}$ or insufficient progress for $\mathcal{F}_t$.
Output: matrix $\mathbf{Q}_t$, pseudo-count vector $\mathbf{b}_t$.

Algorithm 2 Projected subgradient descent to learn a pseudo-count vector $\mathbf{b}$ with a fixed matrix $\mathbf{Q}$.

Input: data $(\mathbf{x_i}, y_i)_i$, a matrix $\mathbf{Q}$, a gradient step size t_0, an initial vector $\mathbf{b}_0$.
Set $t \leftarrow 0$.
Set $\mathbf{b}_t \leftarrow \mathbf{b}_0$.
repeat
  Compute a set $\Omega_t^+ = \{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \mid \mathcal{H}_{ij\ell}(\mathbf{Q}, \mathbf{b}_t) > 0\}$.
  Compute a subgradient $\frac{\partial \mathcal{F}}{\partial \mathbf{b}}$ at $\mathbf{b}_t$ as in Equation (7).
  Calculate $\mathbf{b}_{t+1} \leftarrow \pi_{\mathbb{R}_+^d}\left(\mathbf{b}_t - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{b}}(\mathbf{b}_t)\right)$.
  Calculate $\mathcal{F}_{t+1} \leftarrow \mathcal{F}(\mathbf{Q}, \mathbf{b}_{t+1})$.
  Set $t \leftarrow t + 1$.
until $t < t_{\max}$ or insufficient progress for $\mathcal{F}_t$.
Output: a pseudo-count vector $\mathbf{b}_t$.

where $\circ$ denotes the Schur product between vectors or matrices. Additionally, let Ω_t^+ denote a set of triplet $(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell)$ at iteration t where $\mathcal{H}_{ij\ell}$ is positive at $\mathbf{b}_t$ and a fixed matrix $\mathbf{Q}$,

$$\Omega_t^+ \stackrel{\text{def}}{=} \{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \mid \mathcal{H}_{ij\ell}(\mathbf{Q}, \mathbf{b}_t) > 0\}.$$

Then, we can express a subgradient for the objective function $\mathcal{F}$ at $\mathbf{b}_t$ as follows:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{b}}(\mathbf{b}_t) = \sum_{i,j \rightsquigarrow i} \nabla g(\mathbf{b}_t; \mathbf{x_i}, \mathbf{x_j}) + \mu \sum_{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \in \Omega_t^+} [\nabla g(\mathbf{b}_t; \mathbf{x_i}, \mathbf{x_j}) - \nabla g(\mathbf{b}_t; \mathbf{x_i}, \mathbf{x}_\ell)] + 2\lambda \mathbf{b}_t. \tag{7}$$

Therefore, an update step of the projected subgradient descent for $\mathbf{b}$ with an adaptive monotonic decreasing gradient step size $\frac{t_0}{\sqrt{t}}$ is as follows:

$$\mathbf{b}_{t+1} = \pi_{\mathbb{R}_+^d}\left(\mathbf{b}_t - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{b}}(\mathbf{b}_t)\right),$$

Algorithm 3 Projected subgradient descent with Nesterov's acceleration (PSGD-NES) to learn a matrix $\mathbf{Q}$ and a pseudo-count vector $\mathbf{b}$.

Input: data $(\mathbf{x_i}, y_i)_i$, a gradient step size t_0, an initial matrix $\mathbf{Q}_0$, an initial vector $\mathbf{b}_0$ and a number of periodic iteration τ for updating target neighbors.

Set $t \leftarrow 1$.

Set $\mathbf{b}_{t-1} \leftarrow \mathbf{b}_0$, $\mathbf{b}_{t-2} \leftarrow \mathbf{b}_0$.

Set $\mathbf{Q}_{t-1} \leftarrow \mathbf{Q}_0$, $\mathbf{Q}_{t-2} \leftarrow \mathbf{Q}_0$.

Find target neighbors for each vector $\mathbf{x_i}$ with $d_{\mathfrak{a}}$ as in Equation (5) at $(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1})$.

repeat

Compute $\mathbf{b}_{t-1}^{nes} \leftarrow \pi_{\mathbb{R}_+^d}\Big(\mathbf{b}_{t-1} + \frac{t-2}{t+1}\left(\mathbf{b}_{t-1} - \mathbf{b}_{t-2}\right)\Big)$,

$\mathbf{Q}_{t-1}^{nes} \leftarrow \pi_{\mathcal{S}^+}\Big(\mathbf{Q}_{t-1} + \frac{t-2}{t+1}\left(\mathbf{Q}_{t-1} - \mathbf{Q}_{t-2}\right)\Big)$.

Compute $\Omega^+_{\mathbf{b}_{t-1}} = \left\{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \;\middle|\; \mathcal{H}_{ij\ell}\left(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}\right) > 0\right\}$,

$\Omega^+_{\mathbf{Q}_{t-1}} = \left\{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \;\middle|\; \mathcal{H}_{ij\ell}\left(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}\right) > 0\right\}$.

Compute $\frac{\partial \mathcal{F}}{\partial \mathbf{Q}}\left(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}\right) = \sum_{i,j \rightsquigarrow i} \frac{\partial h(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{Q}}$

$+\mu \sum_{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \in \Omega^+_{\mathbf{Q}_{t-1}}} \left[\frac{\partial h(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{Q}} - \frac{\partial h(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}; \mathbf{x_i}, \mathbf{x}_\ell)}{\partial \mathbf{Q}}\right]$.

$\frac{\partial \mathcal{F}}{\partial \mathbf{b}}\left(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}\right) = \sum_{i,j \rightsquigarrow i} \frac{\partial h(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{b}}$

$+\mu \sum_{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \in \Omega^+_{\mathbf{b}_{t-1}}} \left[\frac{\partial h(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{b}} - \frac{\partial h(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}; \mathbf{x_i}, \mathbf{x}_\ell)}{\partial \mathbf{b}}\right] + 2\lambda \mathbf{b}_{t-1}^{nes}$.

Update $\mathbf{b}_t \leftarrow \pi_{\mathbb{R}_+^d}\Big(\mathbf{b}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{b}}\big(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes}\big)\Big)$, $\mathbf{Q}_t \leftarrow \pi_{\mathcal{S}^+}\Big(\mathbf{Q}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{Q}}\big(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1}\big)\Big)$.

if mod(t, τ) $= 0$ **then**

Update target neighbors for each vector $\mathbf{x_i}$ at $(\mathbf{Q}_t, \mathbf{b}_t)$.

end if

Calculate $\mathcal{F}_t \leftarrow \mathcal{F}(\mathbf{Q}_t, \mathbf{b}_t)$.

Set $t \leftarrow t + 1$.

until $t < t_{\max}$ or insufficient progress for $\mathcal{F}_{t-1}$.

Output: a pseudo-count vector $\mathbf{b}_t$.

where $\pi_{\mathbb{R}_+^d}(\mathbf{x})$ stands for a projection of $\mathbf{x}$ into the strictly positive orthant $\mathbb{R}_+^d$. Since $\mathbb{R}_+^d$ is an open set, $\pi_{\mathbb{R}_+^d}(\mathbf{x})$ is not well-defined. Thus, in practice, we use a minimum positive threshold, $\varepsilon = 10^{-20}$, to implement the projection onto $\mathbb{R}_+^d$. A pseudo-code of the projected subgradient descent to learn pseudo-count vector $\mathbf{b}$ with a fixed matrix $\mathbf{Q}$ is summarized in Algorithm 2.

We have $\mathbf{Q}_0 = \mathbf{P}_0^T \mathbf{P}_0$ where $\mathbf{P}_0$ is a generalized linear transformation matrix of Aitchison embeddings. Therefore, we can consider $\mathbf{P_{alr}}$, $\mathbf{P_{clr}}$ or $\mathbf{P_{ilr}}$ as an initial point for $\mathbf{P}_0$. In term of the initial pseudo-count vector $\mathbf{b}_0$, without any prior knowledge of data, we set $\frac{\mathbf{1}_d}{d}$.

4.3. Projected Subgradient Descent with Nesterov's Acceleration

Although LMNN is an efficient algorithm with a specific purpose solver (Weinberger and Saul, 2008), alternating optimization (Algorithm 1) with LMNN in each iteration seems a

quite high-complexity approach. Thus, we propose a projected subgradient descent with Nesterov's acceleration scheme (Nesterov, 1983) to optimize the parameters $(\mathbf{Q}, \mathbf{b})$ in (6).

Define $h(\mathbf{Q}, \mathbf{b}; \mathbf{x_i}, \mathbf{x_j}) \stackrel{\text{def}}{=} d_{\mathfrak{a}}^2(\mathbf{x_i}, \mathbf{x_j})$ to underline the fact that the distance between $\mathbf{x_i}$ and $\mathbf{x_j}$ is a function of $\mathbf{Q}$ and $\mathbf{b}$. The partial derivatives of h with respect to $\mathbf{Q}$ and $\mathbf{b}$ are

$$\frac{\partial h(\mathbf{Q}, \mathbf{b}; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{Q}} = \left(\log \frac{\mathbf{x_i} + \mathbf{b}}{\mathbf{x_j} + \mathbf{b}}\right)\left(\log \frac{\mathbf{x_i} + \mathbf{b}}{\mathbf{x_j} + \mathbf{b}}\right)^T, \qquad \frac{\partial h(\mathbf{Q}, \mathbf{b}; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{b}} = \nabla g(\mathbf{b}; \mathbf{x_i}, \mathbf{x_j}).$$

At iteration $t+1$, a partial subgradient of $\mathcal{F}$ with respect to $\mathbf{b}$ is given in Equation (7) by setting $\mathbf{Q} = \mathbf{Q}_t$. A subgradient with respect to $\mathbf{Q}$ can be given as

$$\frac{\partial \mathcal{F}}{\partial \mathbf{Q}}(\mathbf{Q}_t, \mathbf{b}_t) = \sum_{i, j \rightsquigarrow i} \frac{\partial h(\mathbf{Q}_t, \mathbf{b}_t; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{Q}} + \mu \sum_{(\mathbf{x_i}, \mathbf{x_j}, \mathbf{x}_\ell) \in \Omega_t^+} \left[\frac{\partial h(\mathbf{Q}_t, \mathbf{b}_t; \mathbf{x_i}, \mathbf{x_j})}{\partial \mathbf{Q}} - \frac{\partial h(\mathbf{Q}_t, \mathbf{b}_t; \mathbf{x_i}, \mathbf{x}_\ell)}{\partial \mathbf{Q}}\right].$$

Nesterov's acceleration scheme, builds gradient updates using a momentum that involves two previous iterations. An update step for $\mathbf{b}$ is given as follows:

$$\mathbf{b}_{t-1}^{nes} = \pi_{\mathbb{R}_+^d}\left(\mathbf{b}_{t-1} + \frac{t-2}{t+1}(\mathbf{b}_{t-1} - \mathbf{b}_{t-2})\right), \qquad \mathbf{b}_t = \pi_{\mathbb{R}_+^d}\left(\mathbf{b}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{b}}(\mathbf{Q}_{t-1}, \mathbf{b}_{t-1}^{nes})\right).$$

while $\mathbf{Q}$ can be updated following

$$\mathbf{Q}_{t-1}^{nes} = \pi_{\mathcal{S}^+}\left(\mathbf{Q}_{t-1} + \frac{t-2}{t+1}(\mathbf{Q}_{t-1} - \mathbf{Q}_{t-2})\right), \qquad \mathbf{Q}_t = \pi_{\mathcal{S}^+}\left(\mathbf{Q}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{Q}}(\mathbf{Q}_{t-1}^{nes}, \mathbf{b}_{t-1})\right).$$

Let $\mathbf{Q} = \mathbf{U}\Sigma\mathbf{U}^T$ by SVD decomposition and $\Sigma_+ \stackrel{\text{def}}{=} \max(\Sigma, \mathbf{0})$, so a projection into the positive semidefinite cone is calculated as $\pi_{\mathcal{S}^+}(\mathbf{Q}) = \mathbf{U}\Sigma_+\mathbf{U}^T$. The proposed projected sugradient descent with Nesterov's acceleration is summarized in Algorithm 3.

4.4. Learning a Low-Rank Generalized Aitchison Embedding

Inspired by the work of Torresani and Lee (2006), we also propose a low-rank learning algorithm to improve the speed of optimizing parameters in the generalized Aitchison embeddings. Instead of learning a psd matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$, we will learn a low-rank matrix $\mathbf{P} \in \mathbb{R}^{m \times d}$, with $m < d$.

It is easy to plug this idea in Algorithm 3 without changing the algorithm structure. In particular, a subgradient of the objective function $\mathcal{F}$ with respect to the matrix $\mathbf{P}$ is as follows:

$$\frac{\partial \mathcal{F}}{\partial \mathbf{P}} = 2\mathbf{P}\frac{\partial \mathcal{F}}{\partial \mathbf{Q}} \in \mathbb{R}^{m \times d}.$$

Since the matrix $\mathbf{P}$ does not need in the positive semidefinite cone, its update step is computed as follows:

$$\mathbf{P}_{t-1}^{nes} = \mathbf{P}_{t-1} + \frac{t-2}{t+1}(\mathbf{P}_{t-1} - \mathbf{P}_{t-2}), \qquad \mathbf{P}_t = \mathbf{P}_{t-1}^{nes} - \frac{t_0}{\sqrt{t}} \frac{\partial \mathcal{F}}{\partial \mathbf{P}}(\mathbf{P}_{t-1}^{nes}).$$

Thus, by replacing a subgradient and an update step in term of $\mathbf{P}$ to those corresponding in $\mathbf{Q}$ in Algorithm 3, we will have a low-rank learning algorithm for parameters in the generalized Aitchison embeddings of the optimization problem (6).

5. Related Work

Notwithstanding Aitchison's work, the logarithm mapping has been consistently applied in information retrieval to correct for the *burstiness* of feature counts (Rennie et al., 2003; Lewis et al., 2004; Madsen et al., 2005). The most common of those mapping is

$$\mathbf{x} \mapsto \log(\mathbf{x} + \alpha \mathbf{1}_d), \tag{8}$$

for $\mathbf{x} \in \mathbb{S}^d$. α is a constant in $\mathbb{R}_+$ which is usually set to $\alpha = 1$ in practice. This embedding can be directly applied to the original histograms or used on term-frequency inverse-document-frequency (TFIDF) transformation and its variants (Aizawa, 2003; Madsen et al., 2005). All of these logarithm maps are particular cases of the embeddings we propose in this work.

In the computer vision literature, the most successful embedding is arguably Hellinger's, which considers the elementwise-square root vector of a histogram ($\mathbf{x} \mapsto \sqrt{\mathbf{x}}$)(Perronnin et al., 2010; Vedaldi and Zisserman, 2012). This embedding was also considered as an adequate representation to learn Mahanlanobis metrics in the probability simplex as argued by Cuturi and Avis (2011, §7.3.2). Some other explicit feature maps such as χ^2, intersection and Jensen-Shannon are also benchmarked in Vedaldi and Zisserman (2012).

6. Experiments

6.1. Setup

We evaluate our algorithms on 12 benchmark datasets of various sizes. Table 1 displays their properties and relevant parameters. These datasets include problems such as scene classification, image classification with a single label or multi labels, handwritten digit and text classification. We follow recommended configurations for these datasets. If they are not provided, we randomly generate 5 folds to evaluate in each run. Additionally, we also repeat the experiments at least 3 times to obtain their averaged results, except for PASCAL VOC 2007 and MirFlickr datasets where we use a predefined training and testing set.

6.2. Metrics and Metric Learning Methods

We consider LMNN metric learning for histograms using: their original representation; the **ilr** representation (Section 2, Equation (3)); Hellinger's map (element-wise square root) of histograms. We also include the simple Euclidean distance in our benchmarks. We expect from a literature survey that *the combination of LMNN and Hellinger's map* to be the best performing of these baselines.

To illustrate the fact that learning the pseudo-count vector $\mathbf{b}$ results in significant performance improvements, we also conduct experiments with an algorithm that learns $\mathbf{Q}$ through LMNN but only considers a uniform pseudo-count vector of $\alpha \in \{0.0001, 0.001, 0.01, 0.1, 1\}$ as in Equation 8. α is selected via cross validation on the training fold. We call this approach `Log-LMNN`.

Table 1: Properties of datasets and their corresponding experimental parameters.

Dataset	#Train	#Test	#Class	Feature	Representation	#Dim	#Run
MIT Scene	800	800	8	SIFT	BoF	800	5
UIUC Scene	1500	1500	15	SIFT	BoF	800	5
DSLR	409	89	31	SURF	BoF	800	5
WEBCAM	646	149	31	SURF	BoF	800	5
AMAZON	2262	551	31	SURF	BoF	800	5
OXFORD Flower	680	680	17	SIFT	BoF	400	5
CALTECH-101	3060	2995	102	SIFT	BoF	400	3
Pascal Voc 2007	5011	4952	20	Dense Hue	BoF	100	1
MirFlickr	12500	12500	38	Dense Hue	BoF	100	1
MNIST	5000	5000	10	Normalized Intensity Level		784	5
20 News Group	600	19397	20	BoW	Topic Modeling	200	5
Reuters	500	9926	10	BoW	Topic Modeling	200	5

6.3. Scene Classification

We conduct experiments on the MIT Scene[1] and UIUC Scene[2] datasets. In these datasets, we select randomly 100 training and 100 testing points from each class. Histograms are obtained by using dense SIFT features with bag-of-feature representation (BoF) where the number of visual words is set to 800. We repeat experiments 5 times on each dataset and split randomly onto training and testing sets.

The two leftmost graphs in Figure 2 shows averaged results with error bars on these datasets. The performance of the proposed embedding improves upon that of LMNN on the original histograms by more than 15% and is slightly better than LMNN combined with the Hellinger map. These graphs also illustrates that Hellinger is the most efficient embedding for histograms. The performance of Hellinger distance is even better than that of LMNN in these datasets. The performances of all alternative embeddings with LMNN are better than those with Euclidean distance respectively.

6.4. Handwritten Digits Classification

We also perform experiments for handwritten digits classification on MNIST[3] dataset. A feature vector for each point is constructed from a normalized intensity level of each pixel. We randomly choose 500 points from each class for training and testing, repeat 5 times for averaged results. The middle graph in Figure 2 illustrates that the generalized Aitchison embedding also outperforms other alternative embeddings.

6.5. Text Classification

We also carry out experiments for text classification on 20 News Groups[4] and Reuters[5] (the 10 largest classes) datasets. In these datasets, we calculate bag of words (BoW) for

1. http://people.csail.mit.edu/torralba/code/spatialenve-lope/
2. http://www.cs.illinois.edu/homes/slazebni/research/
3. http://yann.lecun.com/exdb/mnist/
4. http://qwone.com/~jason/20Newsgroups/
5. http://archive.ics.uci.edu/ml/datasets/Reuters–21578+Text+Categorization+Collection

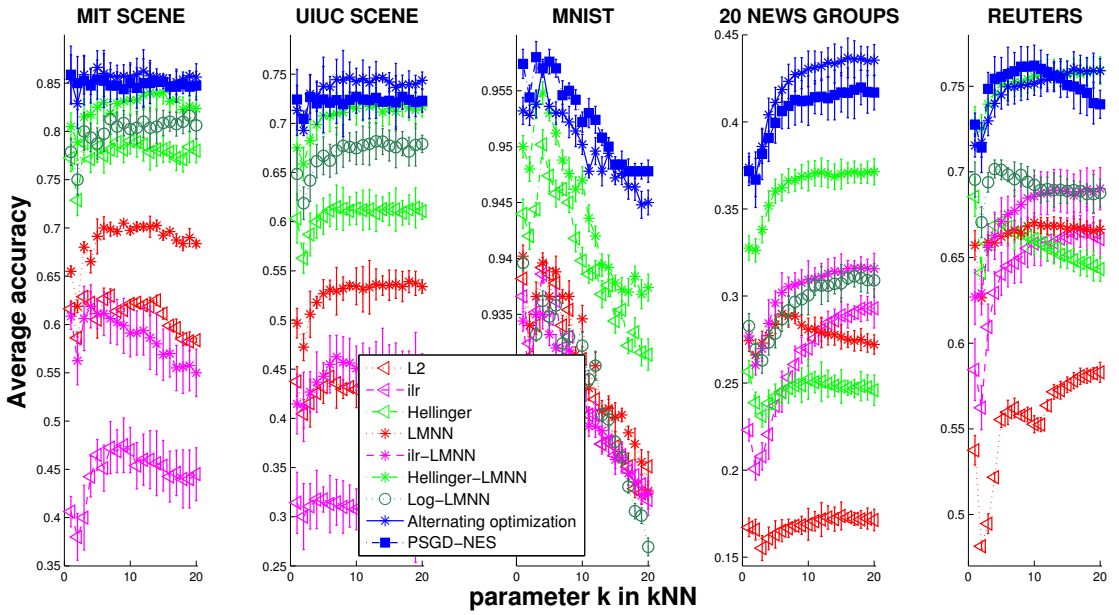

Figure 2: Classification on scene (MIT Scene & UIUC Scene), handwritten digit (MNIST) and text (20 News Group & Reuters).

each document, and then we use topic modeling to reduce the dimension of histograms by *gensim* toolbox[6], so we have a histogram of topics for each document (Blei et al., 2003). We randomly choose 30 points and 50 points from each class in 20 News Groups and Reuters datasets for training respectively, and using the rest for testing. We randomly generate 5 different training and testing sets for each dataset and average results.

The two rightmost graphs in Figure 2 shows that the proposed embedding improves the performance of LMNN on histograms more than 10% on each dataset. It also outperforms original, **ilr** and Hellinger representation on these datasets, except Hellinger representation on Reuters dataset where their performances are comparative.

6.6. Single-label Object Classification

DSLR, AMAZON & WEBCAM We evaluate the proposed embedding on DSLR, AMAZON and WEBCAM datasets[7]. These datasets are widely used for object classification, especially when we considered the performance on various domains of same objects. We randomly split these datasets in 5 folds to evaluate for each run. Each point is a histogram of visual words obtained by BoF representation on SUFR feature where the code-book size is set to 800. We repeat experiments 5 times on each dataset with different random splits and average results.

The three leftmost graphs in Figure 3 illustrate that the performance of the proposed embedding outperforms that of LMNN on these datasets and even improves about 30%, 25% and 10% on DSLR, WEBCAM and AMAZON dataset respectively. Our proposed algorithm also improves the performances of `Log-LMNN` about 7%.

6. http://radimrehurek.com/gensim/
7. http://www1.icsi.berkeley.edu/~saenko/projects.html

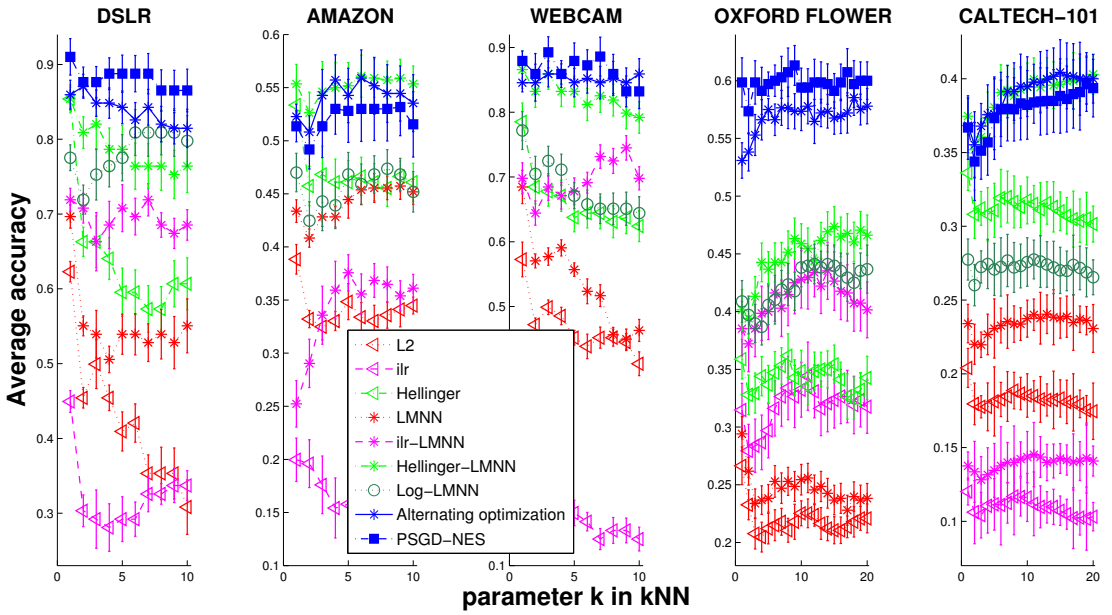

Figure 3: Single-label object classification on DSLR, AMAZON, WEBCAM, OXFORD FLOWER and CALTECH-101.

OXFORD FLOWER We consider the OXFORD Flower[8] dataset. We randomly choose 40 flower images in each class for training and using the rest for testing. We construct histograms by employing BoF representation with 400 visual words on a dense SIFT feature and repeat experiments 5 times on different random splits to obtain averaged results. The fourth graph in Figure 3 shows that the proposed embedding outperforms that of histograms more than 30%, and also improves about 15% comparing to the **ilr** transformation as well as the Hellinger representation with LMNN.

CALTECH-101 We also conduct experiments on CALTECH-101[9] dataset. We randomly choose 30 images for training and up to 50 other images for testing. We use BoF representation with 400 visual words on a dense SIFT feature to construct histograms for each image. The rightmost graph in Figure 3 illustrates averaged results on 3 different random splits of the CALTECH-101 dataset. The proposed embedding appears again as the best choice, outperforms original, **ilr** representation and be comparative to Hellinger's mapping with LMNN.

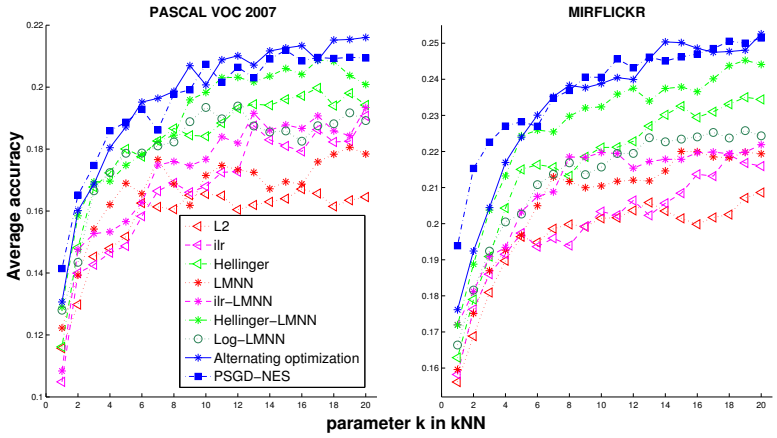

Figure 4: Multi-label object classification on PASCAL VOC 2007 & MirFlickr.

8. http://www.robots.ox.ac.uk/~vgg/data/flowers/17/
9. http://www.vision.caltech.edu/Image_Datasets/Cal-tech101/

6.7. Multi-label Object Classification

We evaluate the proposed method on multi-label image categorization in PASCAL VOC 2007[10] and MirFlickr[11] datasets. We follow a predefined training and testing set for these datasets. Histograms for each image are built in these datasets based on BoF representation with 100 visual words on a dense hue feature. Then, we employ a one-versus-all strategy for k-NN classification and calculate averaged precisions for each dataset. Figure 4 illustrates that the proposed embedding outperforms original, **ilr**, and Hellinger representation with LMNN again. Additionally, the performance of Hellinger distance is better than that of LMNN and comparative with that of `Log-LMNN` in these datasets.

6.8. Low-Rank Generalized Aitchison Embedding

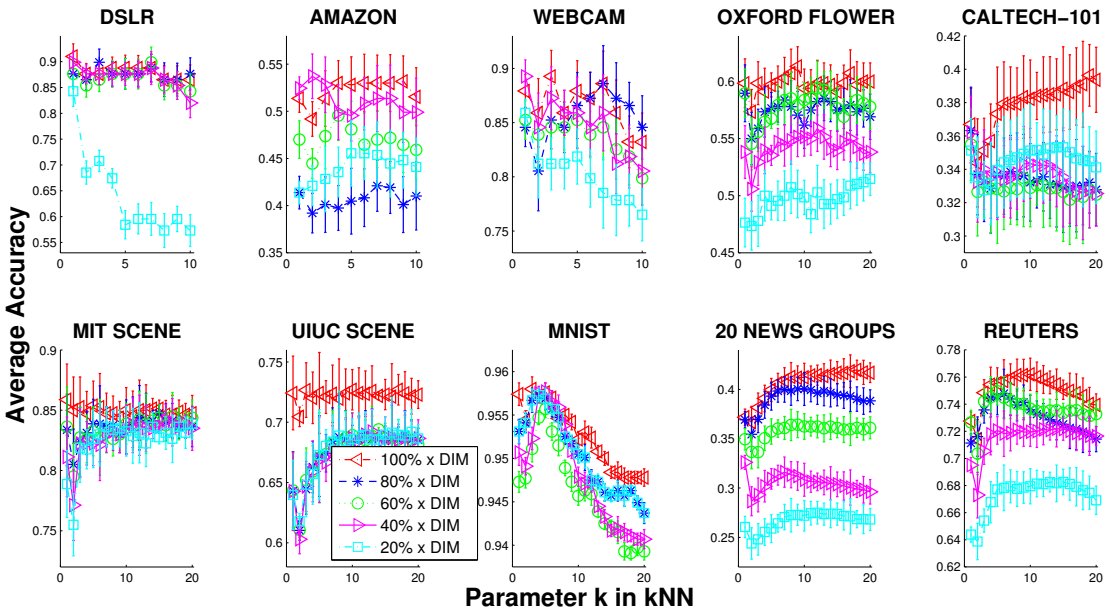

Figure 5: Classification results for low-rank generalized Aitchison embedding.

We conduct experiments for low-rank generalized Aitchison embedding learning where a dimension is set {80%, 60%, 40%, 20%} of the original one in these single-label datasets. Figure 5 illustrates the trade-off performance for low-rank to make the algorithm averaged $\{2\times, 3\times, 4\times, 6\times\}$ faster respectively.

6.9. Computational Speed and Experimental Convergence of the Objective

Figure 6 provides a log-scale time estimate related to an objective value for our proposed alternating optimization (Section 4.2) and projected subgradient descent with Nesterov's acceleration (Section 4.3), also compare with a standard projected subgradient descent. Since we use the LMNN solver, it is only possible to measure time consuming and an objective value for the whole LMNN algorithm instead of those for each iteration. So, there are some gaps in the curve of alternating optimization in Figure 6.

10. http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/
11. http://press.liacs.nl/mirflickr/

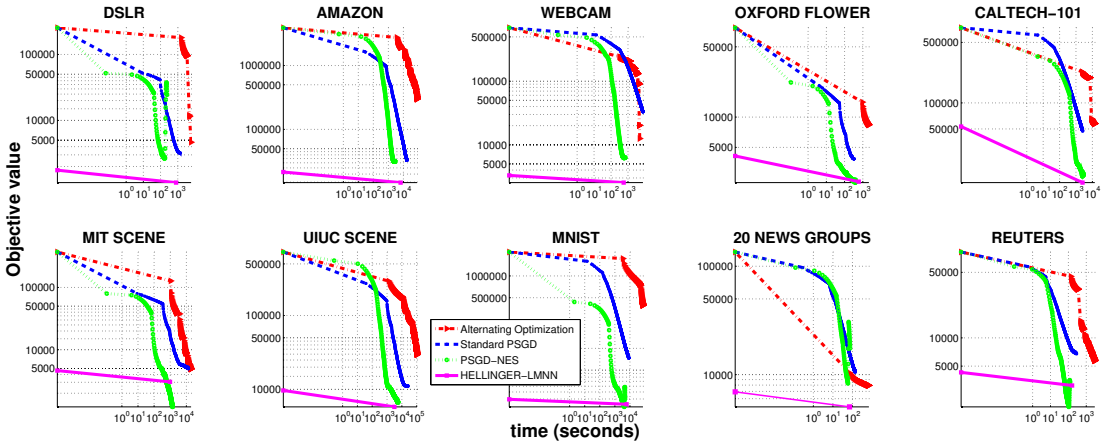

Figure 6: Log-plot illustration for the relation between behavior of the objective function and time consuming in the proposed algorithms and Hellinger-LMNN.

In term of a naive alternating optimization, the running cost is about one order of magnitude larger than that of a direct application of LMNN. This increase in cost is exclusively due to the fact that we run the LMNN solver multiple times. The burden of optimizing the pseudo-count vector is small due to the fact that the gradient has a closed-form solution for each pair in the objective function. We only need to run a few iterations of the LMNN algorithm using warm start when alternating. Our experiments show that we only need to run 6 to 10 alternating iterations for these datasets. However, its time consuming seems still quite high. Therefore, we also propose the projected subgradient descent with Nesterov's acceleration to significantly reduce time consuming as showed in Figure 6 while its performances are comparative with the alternating optimization approach as illustrated in Figure 2, Figure 3 and Figure 4.

7. Conclusion

We illustrate that the proposed embedding is an effective representation for histograms, it outperforms histograms in the original, **ilr** and Hellinger substantially from scene, object with a single label or multi labels to handwritten digit and text classification. We also show that our jointly learning parameters for the generalized Aitchison embedding algorithm substantially improve the performance of the algorithm which only learns Mahalanobis matrix by LMNN and uses a uniform pseudo-count vector chosen via cross-validation (`Log-LMNN`). Especially, the proposed projected sugradient descent with Nesterov's acceleration is efficient in both performance and time consuming which covers the main drawback about computational burden of the naive alternating optimization. Additionally, the proposed low-rank method also helps improve computational aspect for the generalized Aitchison embedding approach more effectively.

Acknowledgments

We thank anonymous reviewers for their comments. TL acknowledges the support of the MEXT scholarship 123353. MC acknowledges the support of the Japanese Society for the Promotion of Science grant 25540100.

References

J. Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society*, 44:139–177, 1982.

J. Aitchison. *The statistical analysis of compositional data*. Chapman & Hall, Ltd., 1986.

J. Aitchison. A concise guide to compositional data analysis. In *CDA Workshop*, 2003.

J. Aitchison and I. J. Lauder. Kernel density estimation for compositional data. *Applied statistics*, pages 129–137, 1985.

J. Aitchison and S. M. Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, pages 261–272, 1980.

A. Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, pages 147–154, 2006.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

C. Burge, A.M. Campbell, and S. Karlin. Over-and under-representation of short oligonucleotides in dna sequences. *National Academy of Sciences*, 89(4):1358–1362, 1992.

W.M. Campbell, J.P. Campbell, D.A. Reynolds, D.A. Jones, and T.R. Leek. Phonetic speaker recognition with support vector machines. In *Advances in Neural Information Processing Systems*, 2003.

M. Cuturi and D. Avis. Ground metric learning. *arXiv preprint arXiv:1110.2306*, 2011.

J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning*, pages 209–216, 2007.

G. Doddington. Speaker recognition based on idiolectal differences between speakers. In *Eurospeech*, pages 2521–2524, 2001.

J. J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barcel-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3): 279–300, 2003.

A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, pages 451–458, 2005.

J. Goldberger, S. T. Roweis, G. E. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*, 2004.

T. Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Springer, 2002.

B. Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 1981.

D. Kedem, S. Tyree, K. Q. Weinberger, F. Sha, and G. Lanckriet. Nonlinear metric learning. In *Advances in Neural Information Processing Systems*, pages 2582–2590, 2012.

J. T. Kwok and I. W. Tsang. Learning with idealized kernels. In *International Conference on Machine Learning*, pages 400–407, 2003.

C. S. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. In *Pacif. Symp. on Biol.*, volume 7, pages 566–575, 2002.

D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *International Conference on Machine Learning*, pages 545–552, 2005.

Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *Computer Vision & Pattern Recogition*, pages 2297–2304, 2010.

J. D Rennie, L. Shih, J. Teevan, and D. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Int. Conf. on Machine Learning*, volume 3, pages 616–623, 2003.

G. Salton and M. J. McGill. *Introduction to Moderm Information Retrieval*. McGraw-Hill, 1983.

M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems*, volume 16, page 41, 2003.

S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. Online and batch learning of pseudo-metrics. In *International Conference on Machine Learning*, page 94, 2004.

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems*, pages 1385–1392, 2006.

A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Pattern Analysis and Machine Intelligence*, 34(3):480–492, 2012.

K.Q. Weinberger and L.K. Saul. Fast solvers and efficient implementations for distance metric learning. In *Int. Conf. on Machine Learning*, pages 1160–1167, 2008.

K.Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Adv. Neural Inf. Process. Syst.*, pages 1473–1480, 2006.

E. P. Xing, A. Y. Ng, M. I. Jordan, and S. J. Russell. Distance metric learning with application to clustering with side-information. In *Adv. Neural Inf. Process. Syst.*, pages 1473–1480, 2002.